Reconfigurable Hardware Security

Ryan Kastner

Department of Electrical and Computer Engineering

University of California, Santa Barbara

CISR Lecture

Naval Postgraduate School

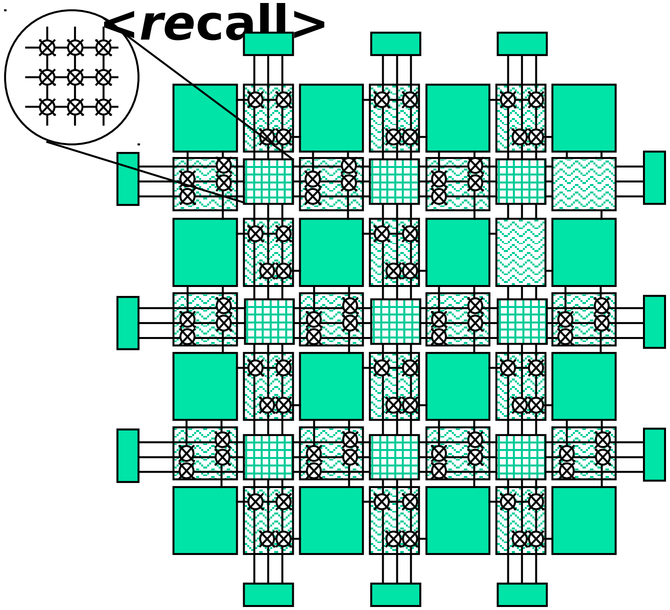
August 2005

Outline

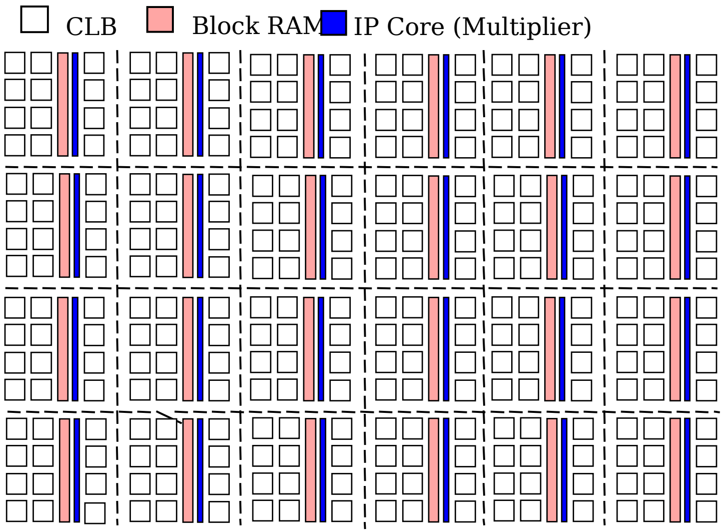
- Reconfigurable Hardware
  - Brief History of Reconfigurable Computing
  - Benefits
  - FPGA Architectures
  - Configuration/Programming
- Security Issues
  - General Hardware Security
  - FPGA Attacks
  - FPGA Virus
  - FGPA Security

Reconfigurable Hardware

Main Entry: re-
Function: *prefix*
1 : again : anew
<*retell*>
2 : back : backward
<*recall*>



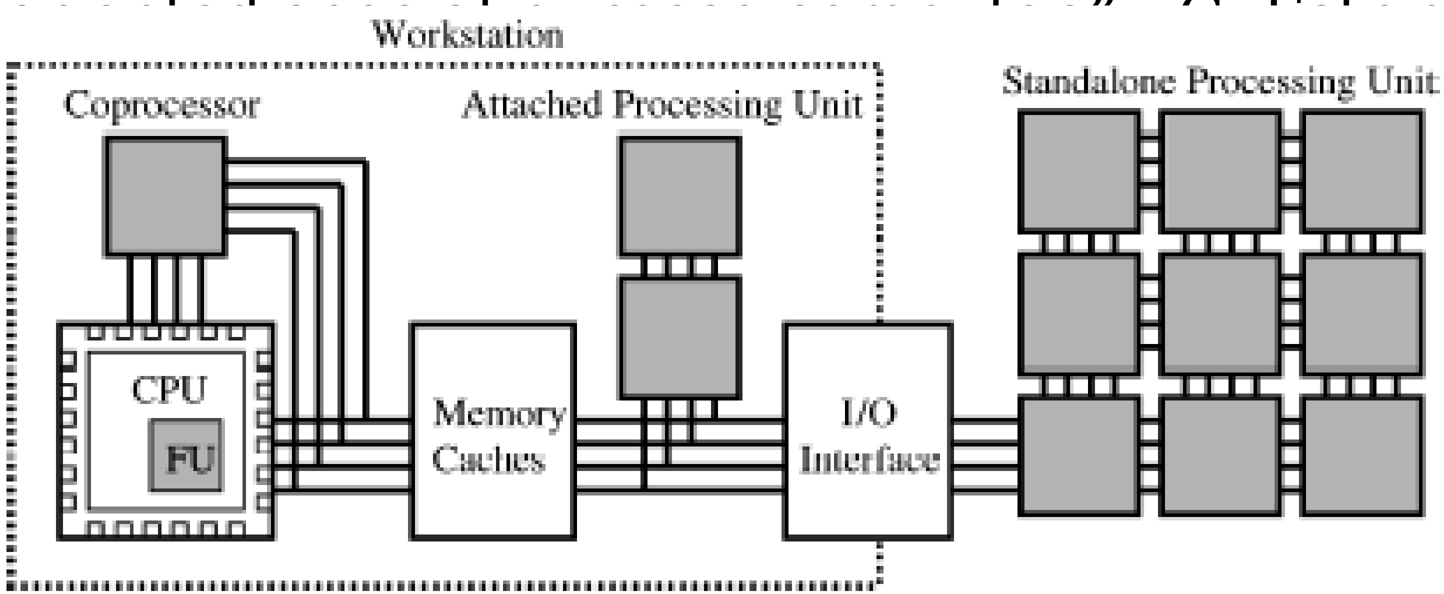
Main Entry: con·fig·ure
Pronunciation: k&n-'fi-gy&r
Function: transitive verb
: to set up for operation especially in a particular way



KEY ADVANTAGE: Performance of Hardware, Flexibility of Software

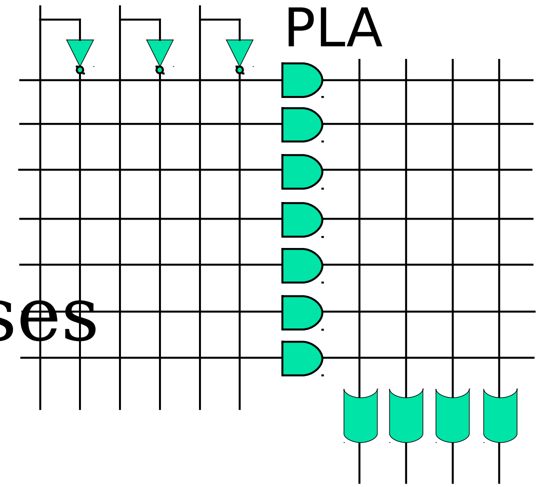
Origins of Reconfigurable Computing

- Fixed-Plus-Variable (F+V) Structure Computer
  - Standard processor augmented with inventory of reconfigurable building blocks
  - "… to permit computations which are beyond the capabilities of present systems by providing an inventory of high speed substructures and rules for interconnecting them such that the entire system may be temporarily distorted into a problem

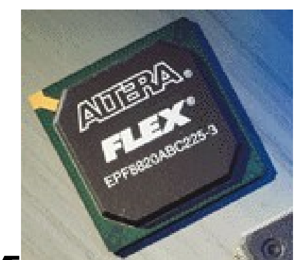


History of Reconfigurable Computing

- Early Years – PLA, PAL
  - One time programmable
  - Programmed by blowing fuses



- Complex Logic Devices
  - FPGA, CPLD
  - Reprogrammable (SRAM)
  - Initially used for glue logic, ASIC prototyping



"Moore" transistors = more complex devic

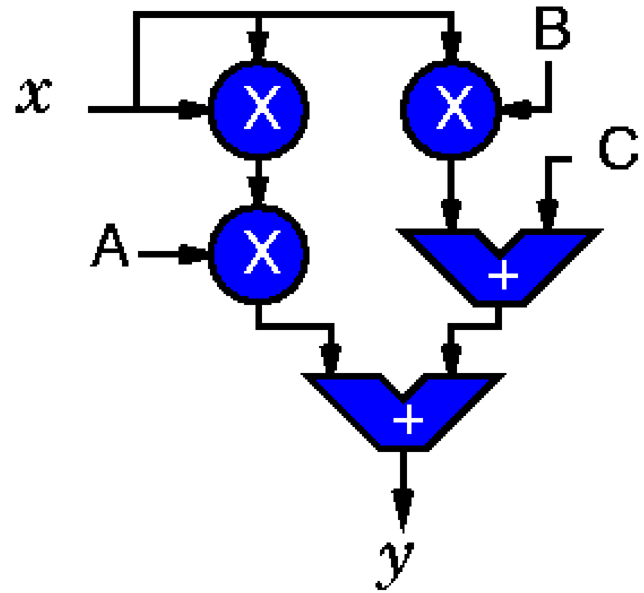
Modern Reconfigurable Devices

- Reconfigurable devices are extremely complicated, multiprocessing computing systems
  - Mix of hardware and software components
    - Microprocessors – RISC, DSP, network, …
    - Logic level (FPGA) Reconfigurable logic
  - Specs for Xilinx Virtex II
    - 3K to 125K logic cells,
    - Four PowerPC processor cores
    - Complex memory hierarchy - 1,738 KB block RAM, external memory, local memory in CLBs
    - Possibility of soft core processors – DSP
    - Custom hardware - embedded multipliers, fast carry chain logic, etc.

- Can implement complex applications

Traditional Choice: Hardware vs. Software

- Properties of Hardware
  - Fast: High performance $y = Ax^2 + Bx + C$
    - Spatial execution
    - Adaptable parallelism
  - Compact: Silicon area efficient
    - Operations tailored to application
    - Simple control
    - Direct wire connections between operations



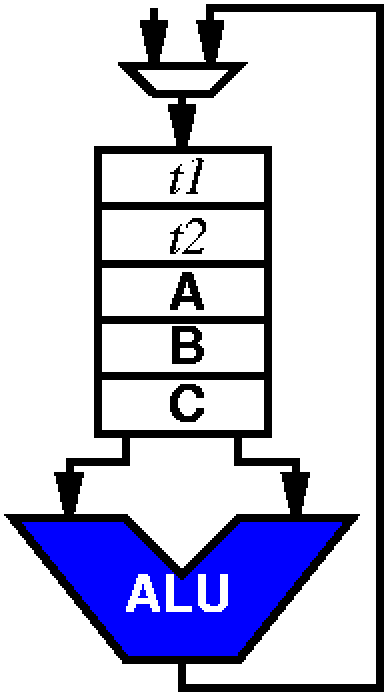
Source: DeHon/Wawrzynek

Hardware Inflexible: Fixed at Fabrication

Traditional Choice: Hardware vs. Software

- Properties of Software
  - Slow:
    - Sequential execution
    - Overhead of interpreting instructions
  - Area inefficient:
    - Fixed width, general operators
    - Area overhead
      - Instruction cache
      - Control circuitry

$$t1 \leftarrow x$$
$$t2 \leftarrow \mathbf{A} \times t1$$
$$t2 \leftarrow t2 + \mathbf{B}$$
$$t2 \leftarrow \mathbf{t2} \times t1$$
$$y \leftarrow \mathbf{t2} + \mathbf{C}$$



Souce: DeHon/Wawrzynek

Software Flexible: Fixed at Runtime

Reconfigurable Hardware

- Fast:
  - Spatial parallelism (like hardware)
  - Application specific operators, control circuitry
- Flexible:
  - Operators and interconnect

[redacted]

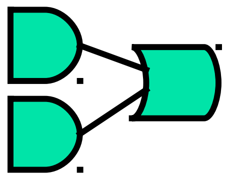
Reconfigurable Hardware

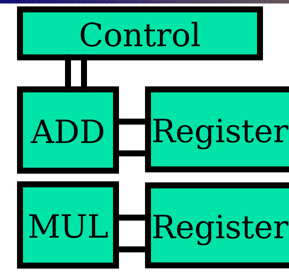
- Fast and flexible, but…
  - Area overhead – switches, configuration bits
  - Delay overhead – switches, logic
  - Compilation
    - Difficult - architecture “too” flexible
    - [redacted]

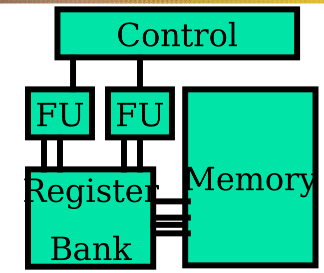
Performance Benefits

- Up to 100x performance increase compared to processors (microprocessor, DSP)
- ~10x performance density advantage over processors
- Applications like:
  - Pattern matching
  - Data encryption
  - Data compression
  - Digital communications
  - Video and image processing
  - Boolean satisfiability
  - Networking
  - Cryptography

Classification of Reconfigurable Architectures







	Logic Level	**Instruction Level**	**Function Level**
Programming Unit	**Bit**	**Byte**	**Operands**
Basic Unit of Computation	**Boolean Operation (and, or, xor)**	**Arithmetic Operation**	**Functional Operation**
Communication	**Wires, Flip Flops**	**Bundles of Wires, Registers**	**Bus, Memory**
Example Devices	**FPGA, CPLD**	**PRISC, Chimaera, Garp**	**NAPA, RAW, RaPiD**

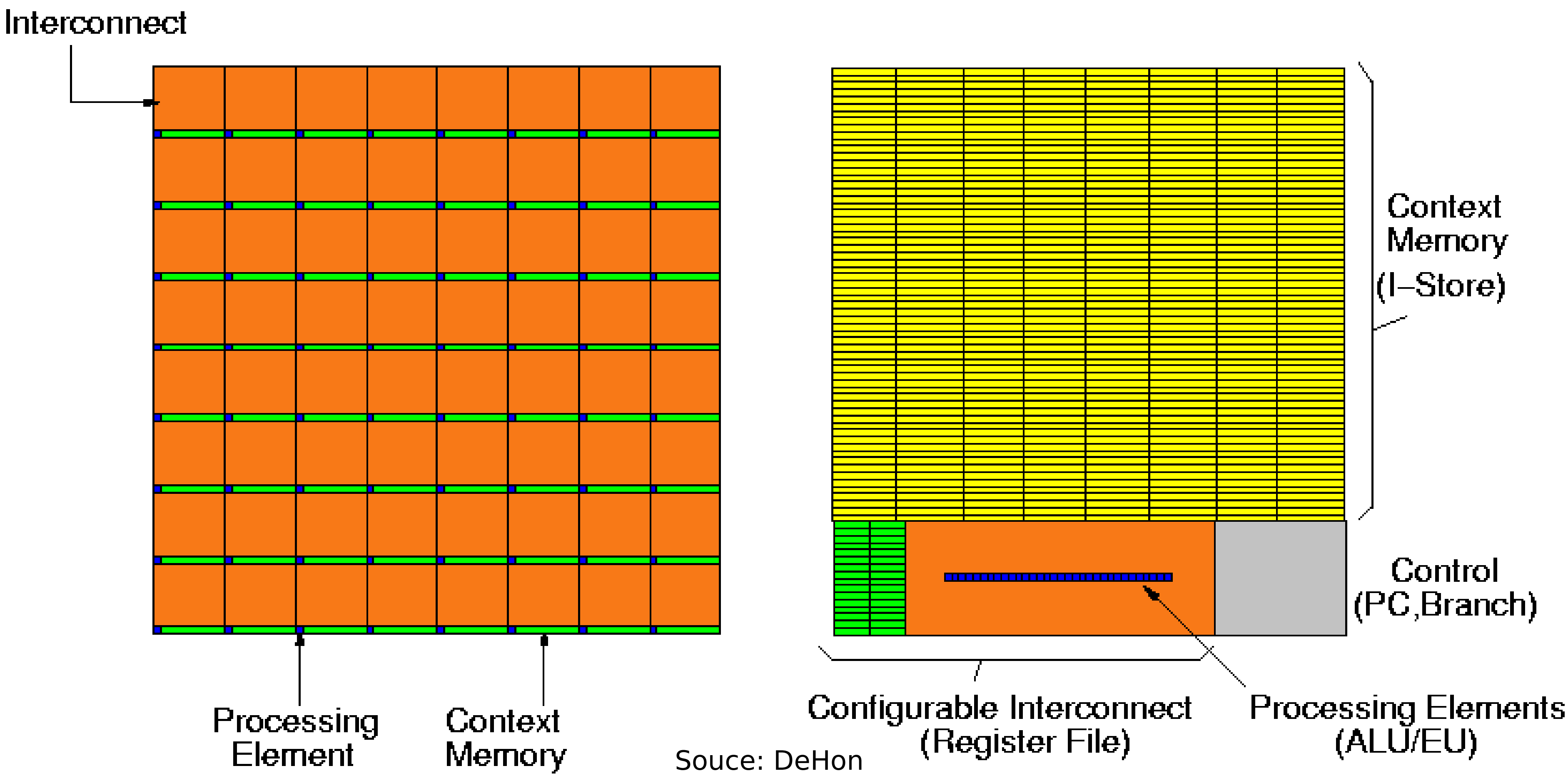
Flexibility ←

Performance →

Power/Energy Consumption ←

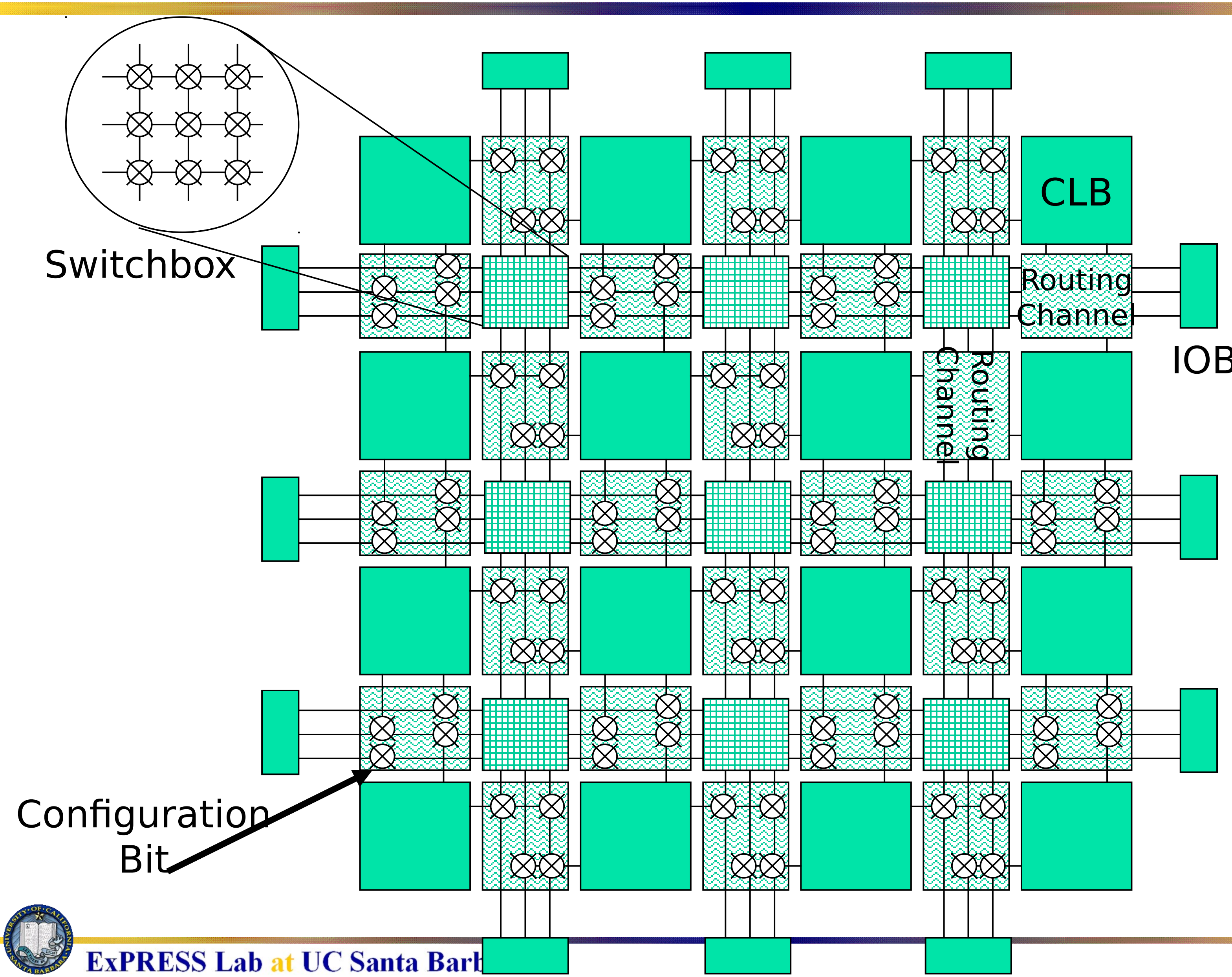
Design Complexity ←

Processor vs FPGA



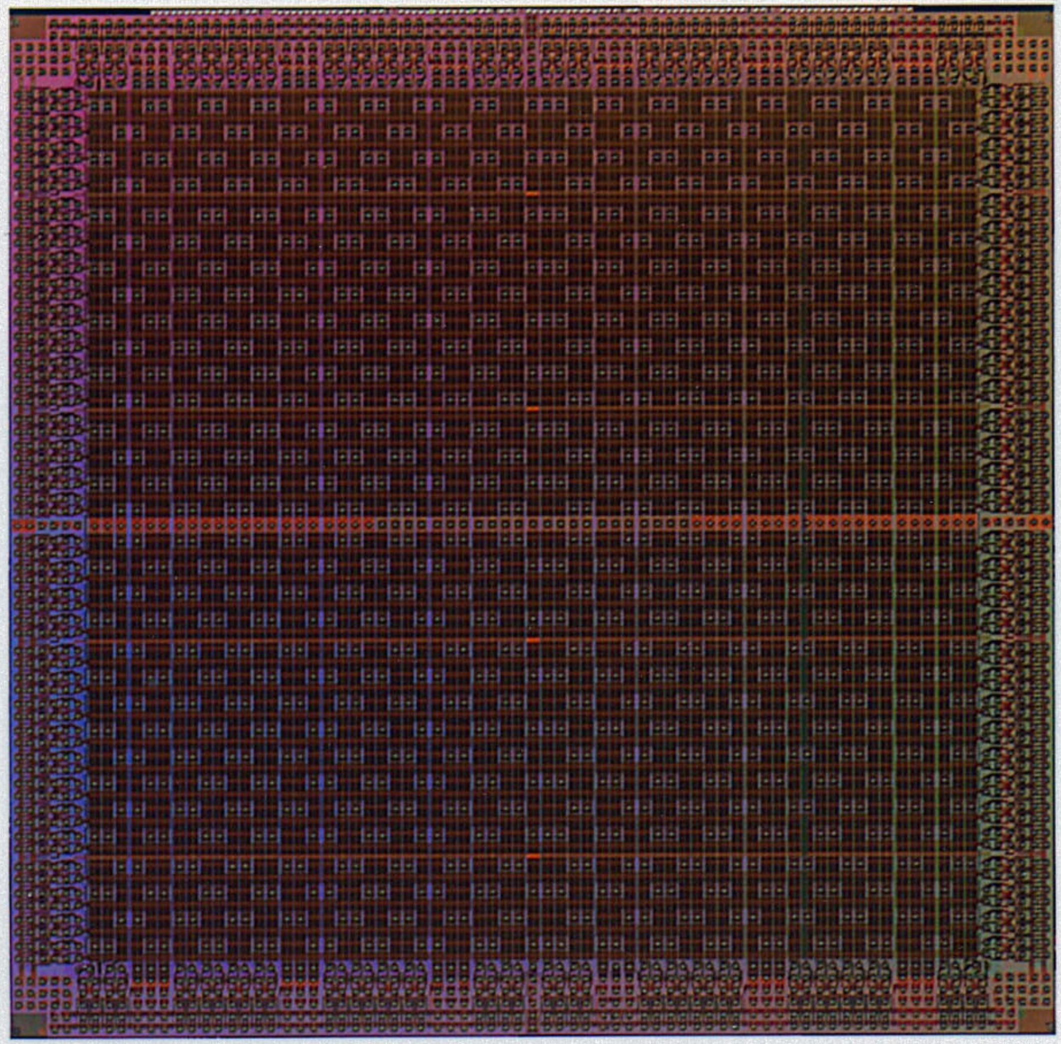
Souce: DeHon

FPGA



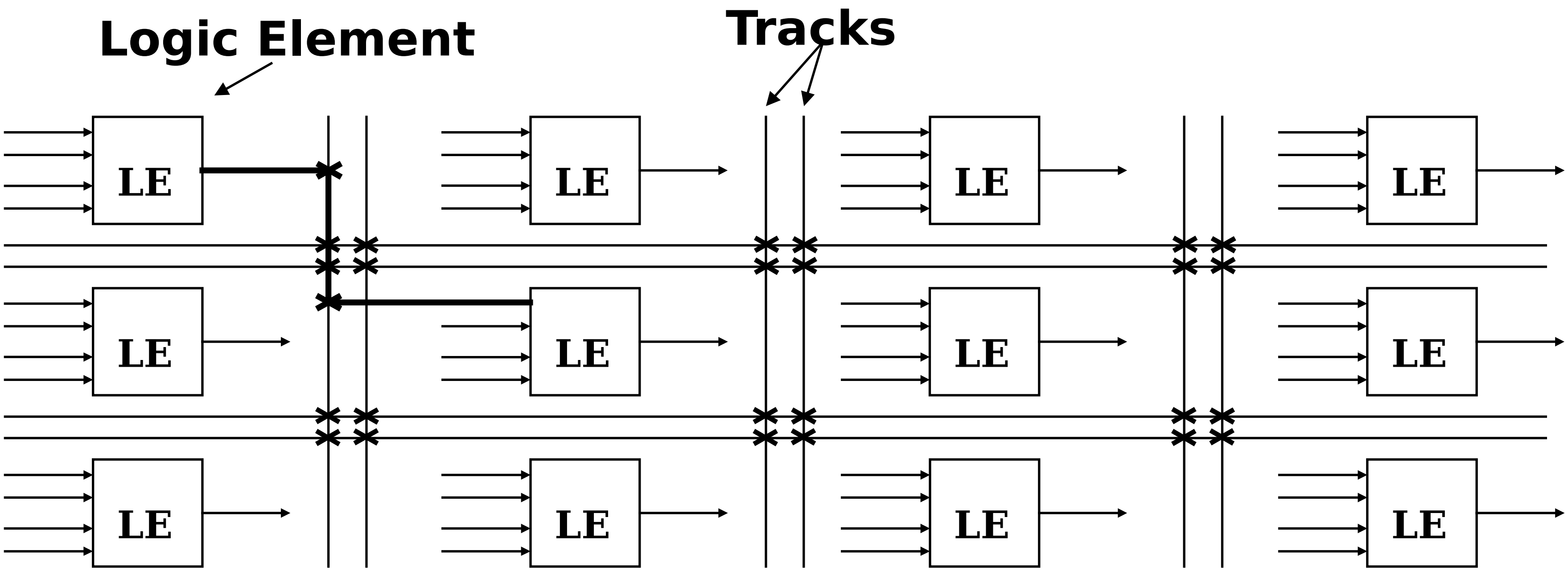
Switchbox
CLB
Routing Channel
IOB
Routing Channel
Configuration Bit

FPGA



Lattice first FPGA device on Fujitsu 130nm technology

Programmable Logic



- Each *logic element* outputs one data bit
- Interconnect programmable between elements
- Interconnect *tracks* grouped into channels

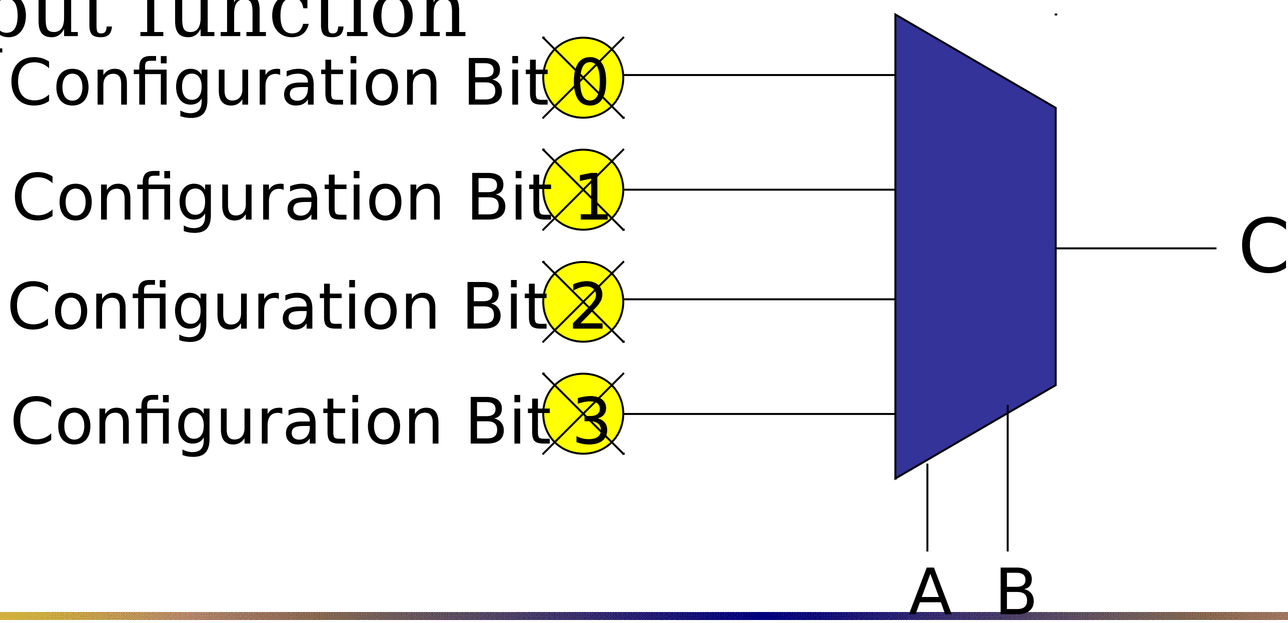
Lookup Table (LUT)

- Program configuration bits for required functionality
- Computes “any” 2-input function

2-LUT

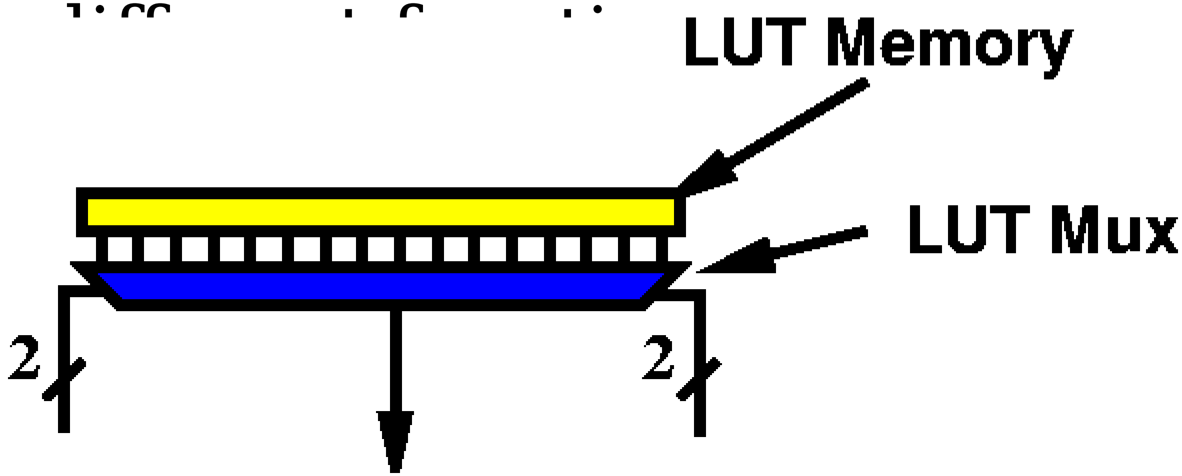
In	Out
00	0
01	0
10	0
11	1

A, **B** → **C=A · B**



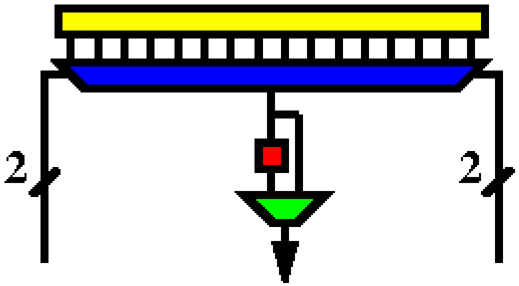
Lookup Table (LUT)

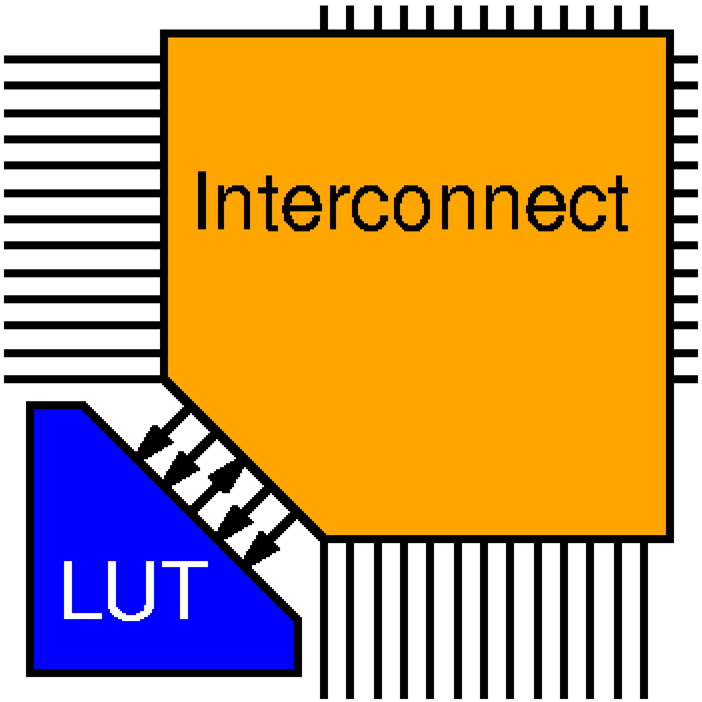
- K-LUT -- K input lookup table
- Any function of K inputs by programming table
- Load bits into table
  - 2^N bits to describe functions
  - => 2^{2^N} different functions



Lookup Table (LUT)

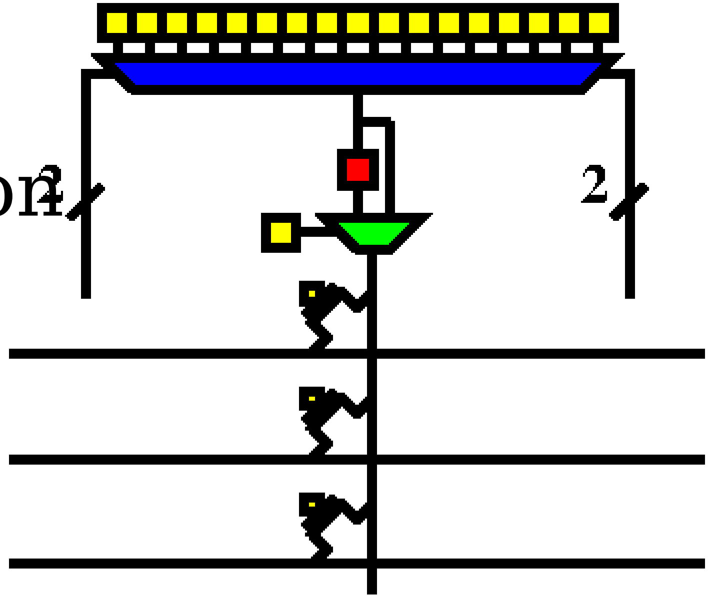
K-LUT (typical k=4)
w/ optional
output Flip-Flop



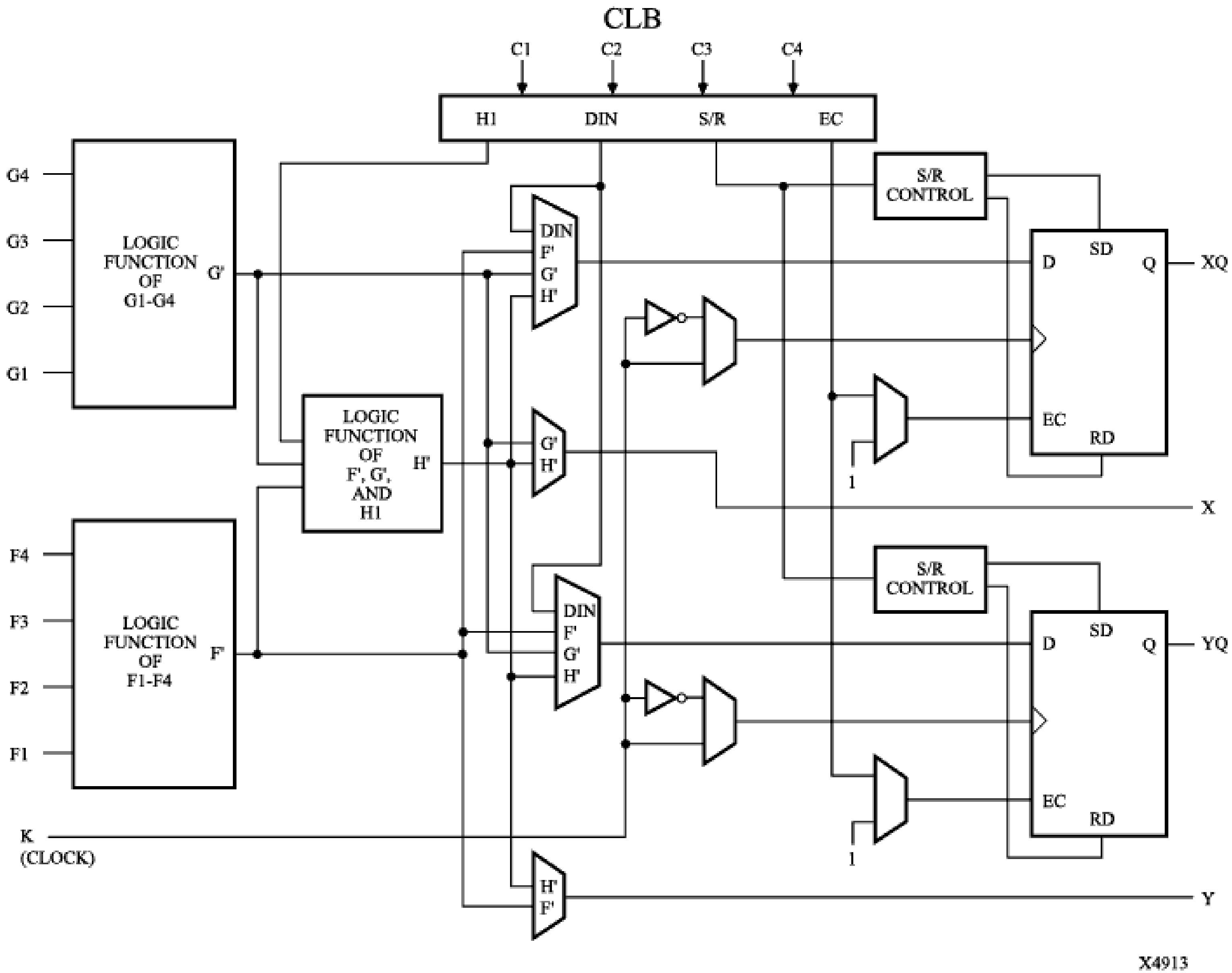


Lookup Table (LUT)

- Single configuration bit for each:
  - LUT bit
  - Interconnect point/option
  - Flip-flop select



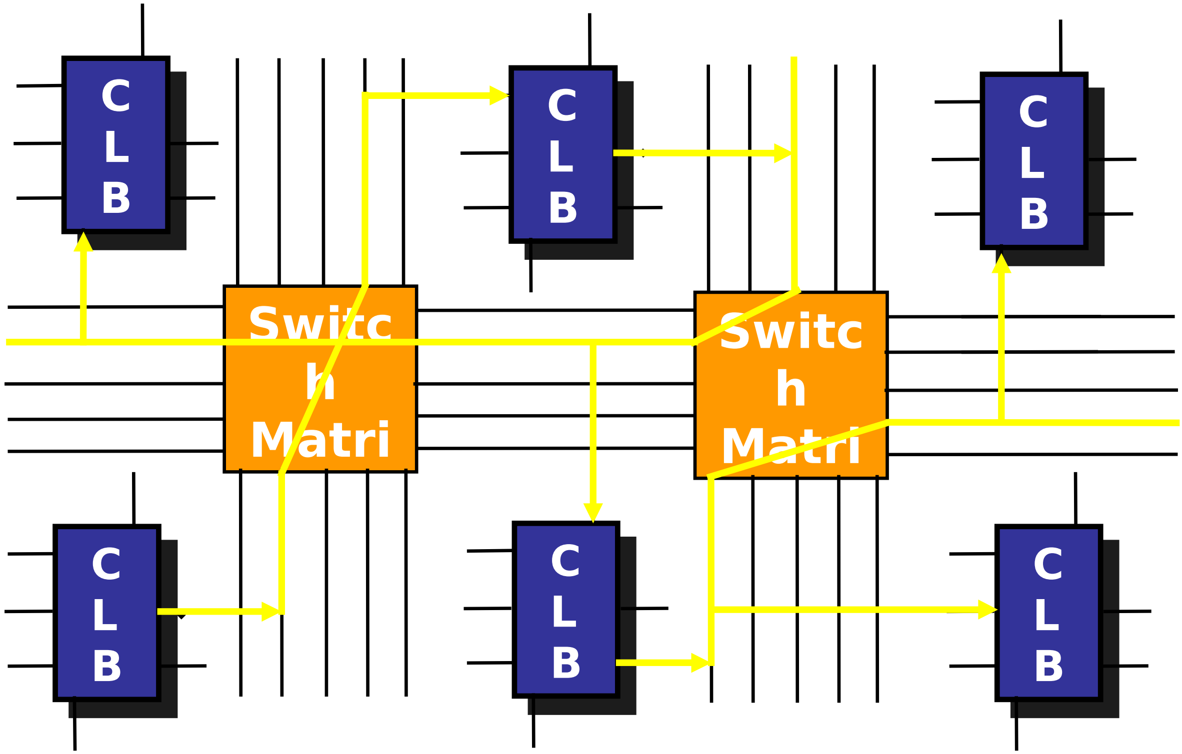
Configurable Logic Block (CLB)



Ex

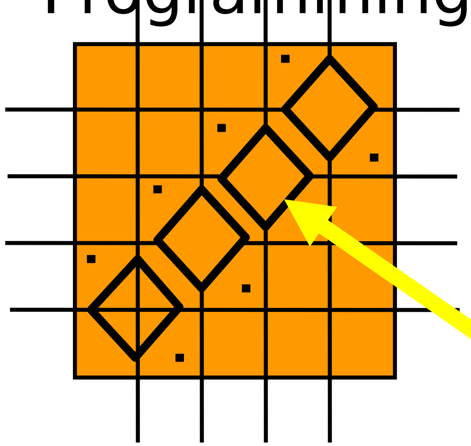
Programmable Interconnect

- Interconnect architecture
  - Fast local interconnect
  - Horizontal and vertical lines of various lengths

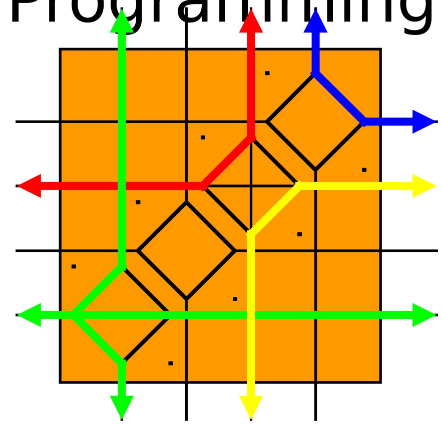


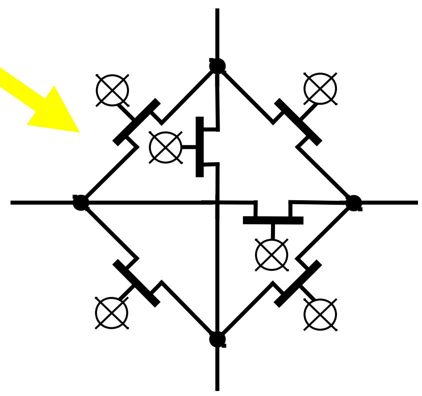
Switchbox Operation

Before Programming



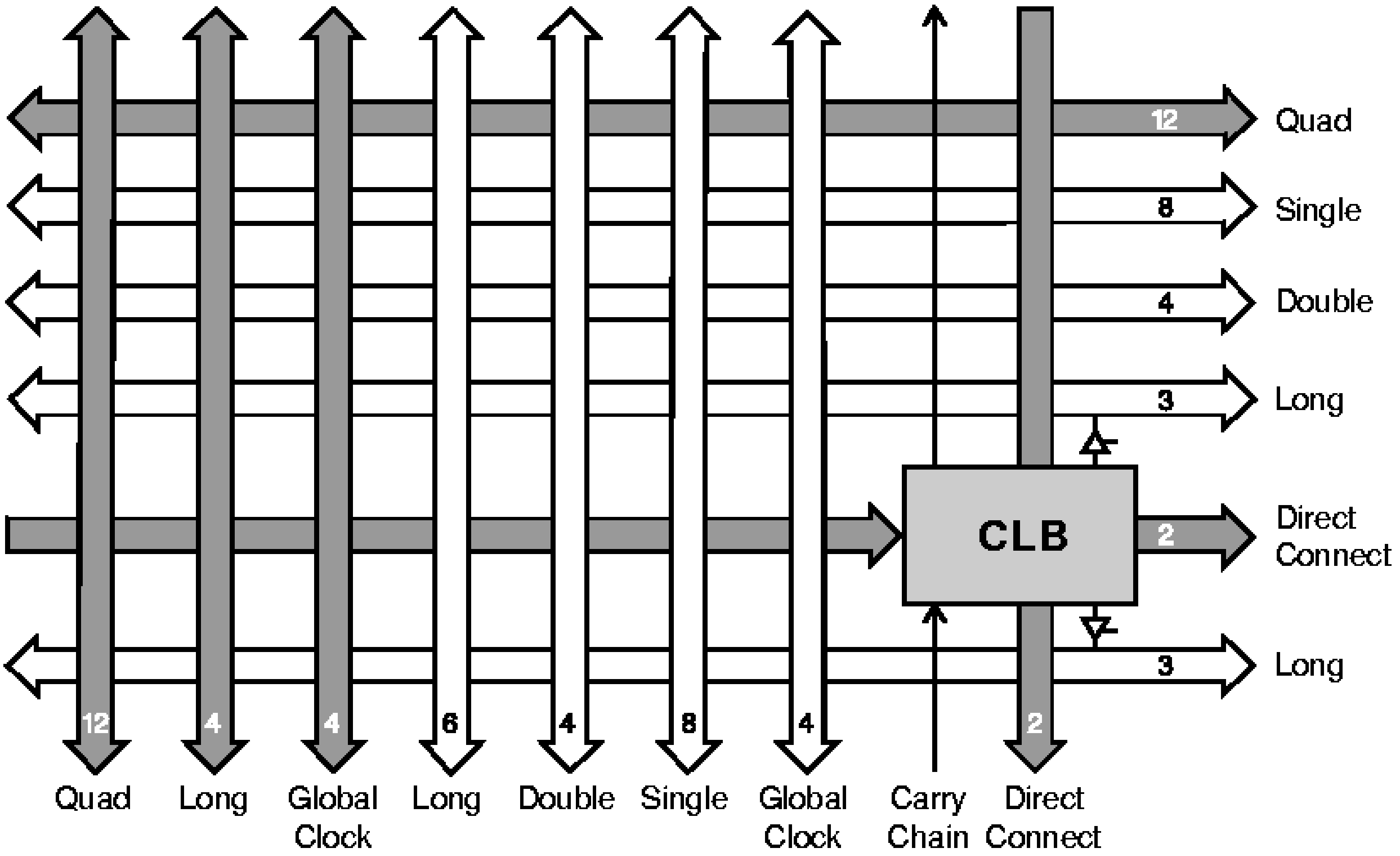
After Programming



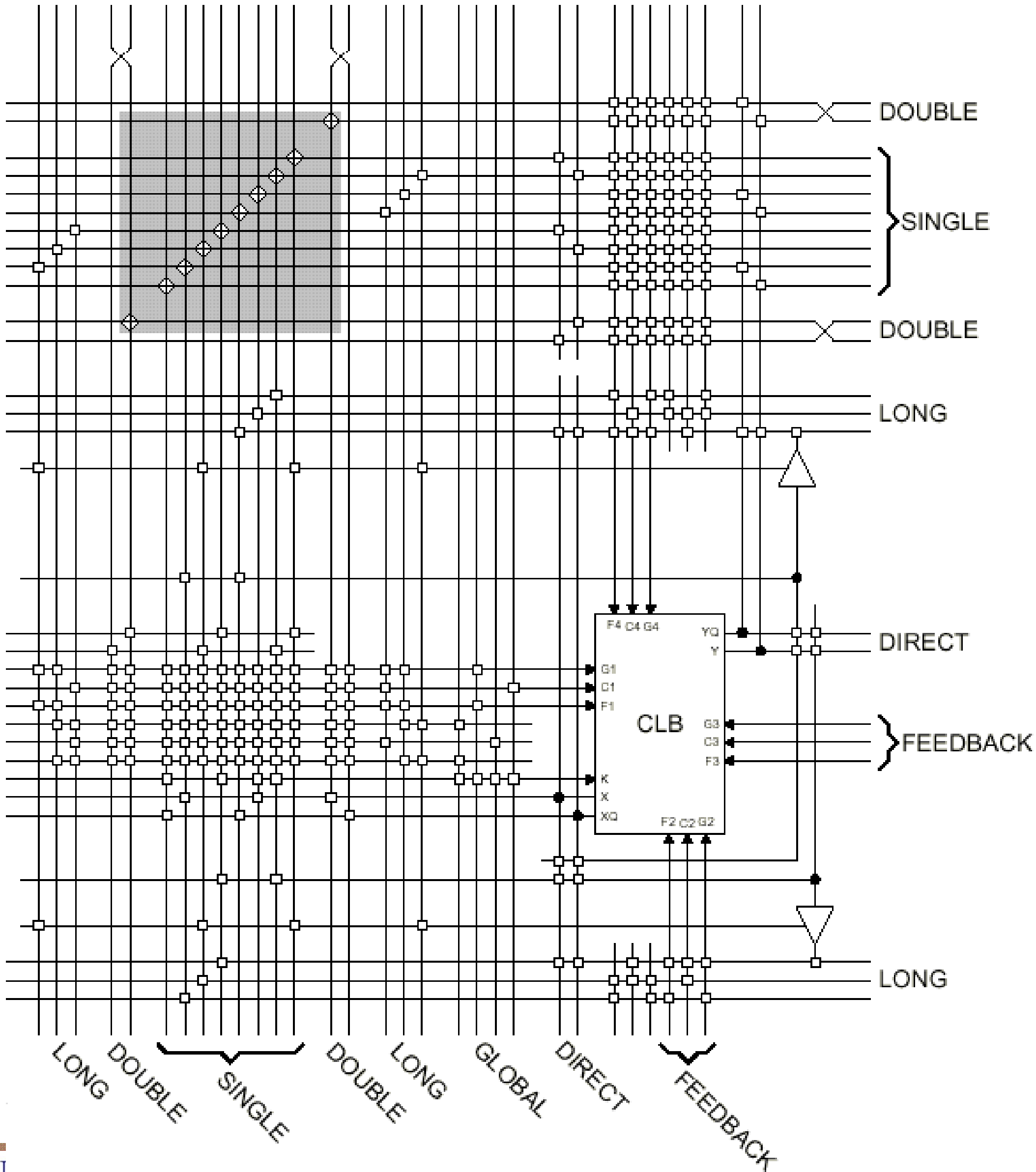


- 6 pass transistors per switchbox interconnect point
- Pass transistors act as programmable switches
- Pass transistor gates are driven by configuration memory cells

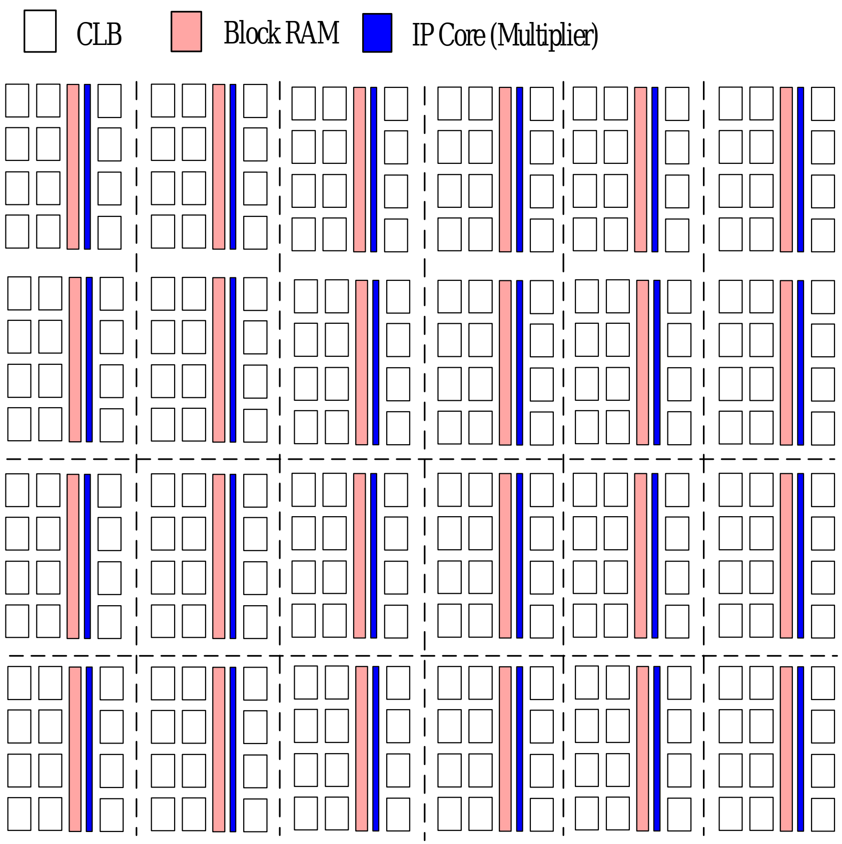
Programmable Interconnect



Programmable Interconnect



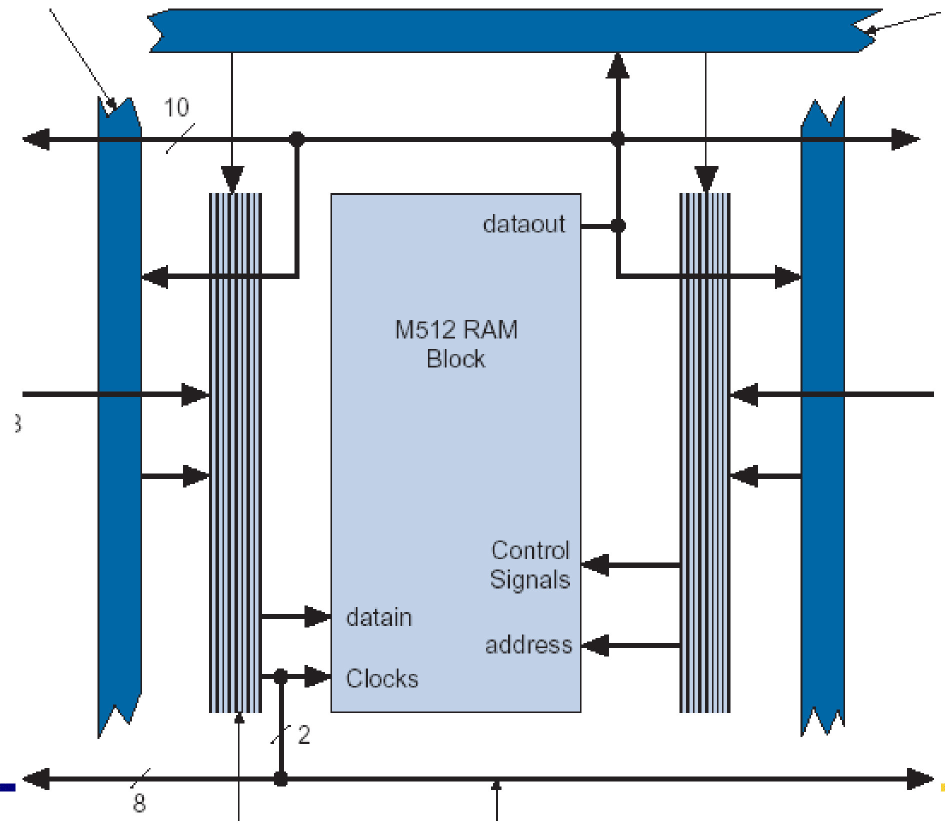
Embedded Functional Units



- Fixed, fast multipliers
- MAC, Shifters, counters
- Hard/soft processor cores
  - PowerPC
  - Nios
  - Microblaze
- Memory
  - Block RAM
  - Various sizes and distributions

Embedded RAM

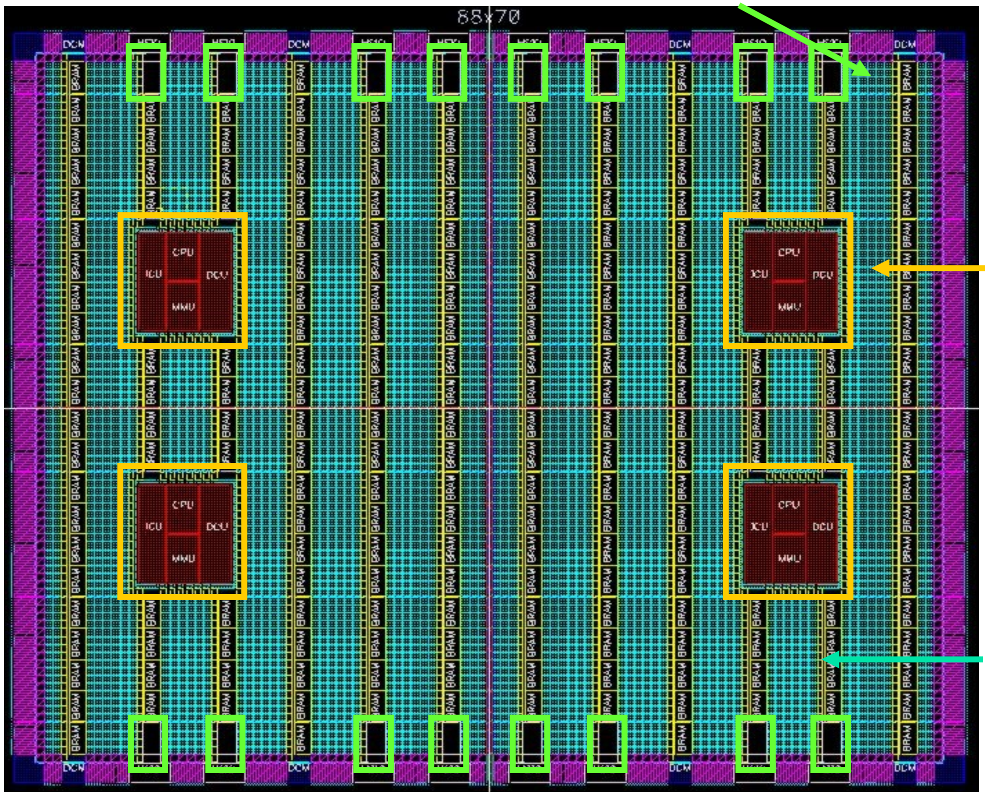
- Xilinx – Block SelectRAM
  - 18Kb dual-port RAM arranged in columns
- Altera – TriMatrix Dual-Port RAM
  - M512 – 512 x 1
  - M4K – 4096 x 1
  - M-RAM – 64K x 8



Xilinx Virtex-II Pro

- 1 to 4 PowerPCs
- 4 to 16 multi-gigabit transceivers
- 12 to 216 multipliers
- 3,000 to 50,000 logic cells
- 200k to 4M bits RAM
- 204 to 852 I/Os

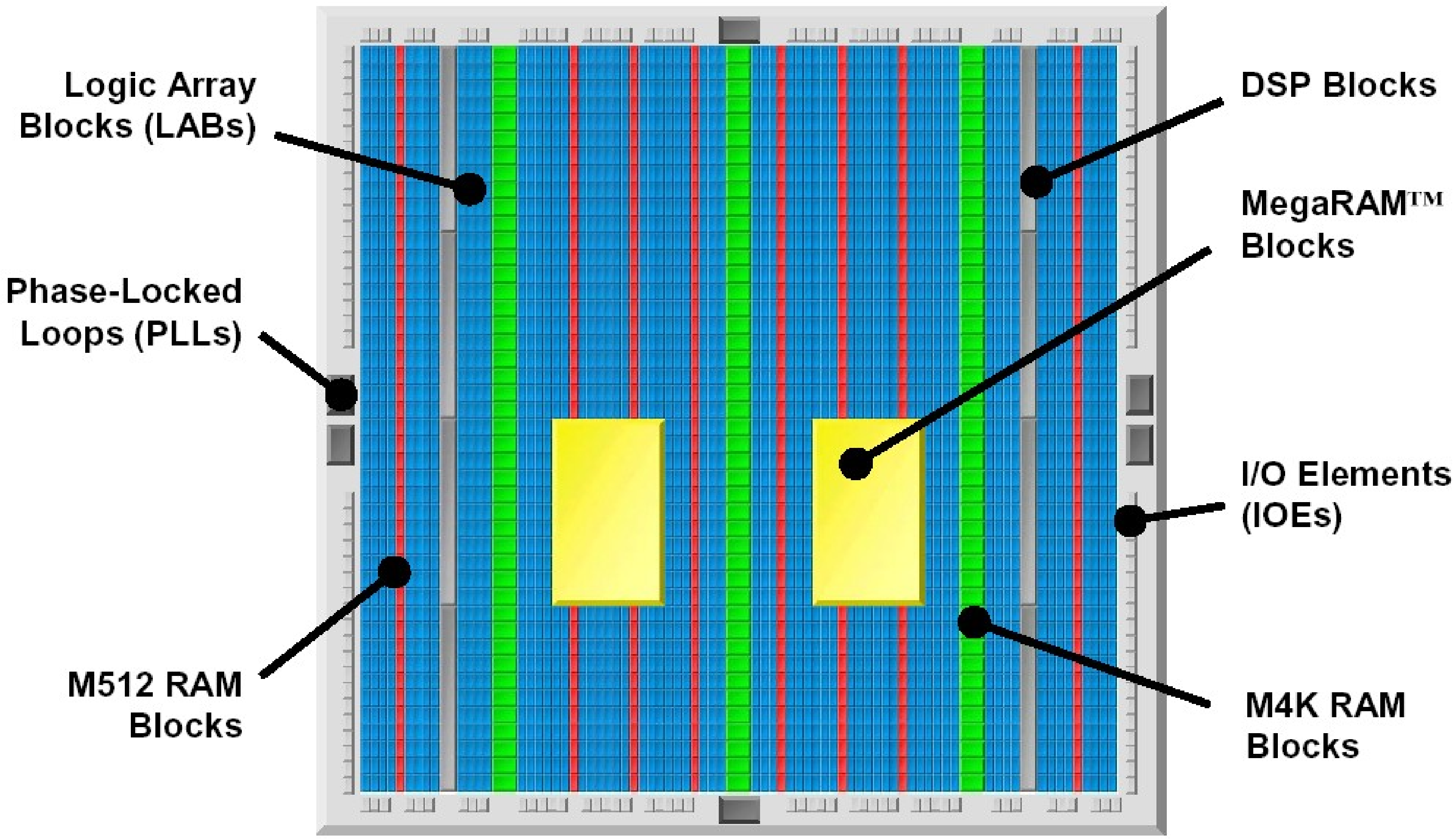
Up to 16 serial transceivers
- 622 Mbps to 3.125 Gbps



PowerPCs

Logic cells

Altera Stratix

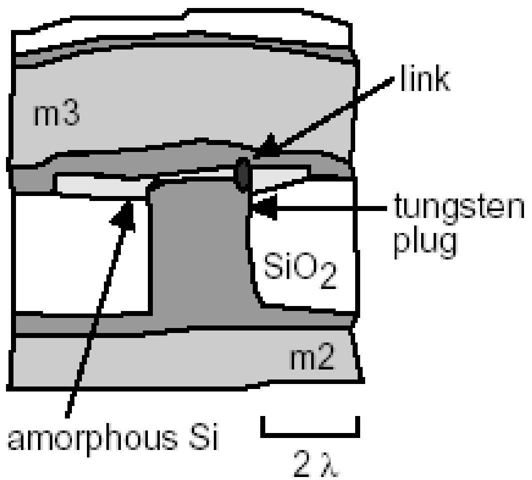


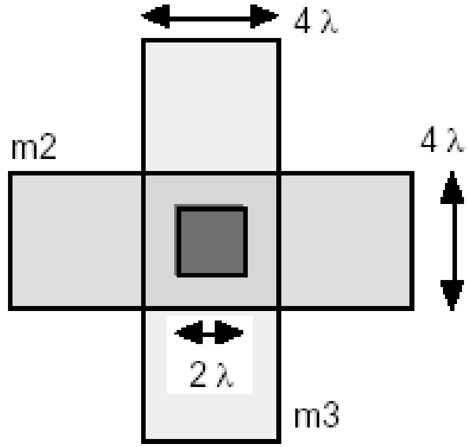
Programming the FPGA

- Configure logic – CLBs, LUTs, FFs
- Configure interconnect – channel, switchbox
- Large number of bits – 10s MB
- Configuration bit technology
  - Antifuse (program once)
  - SRAM
  - Floating Gate (Flash)

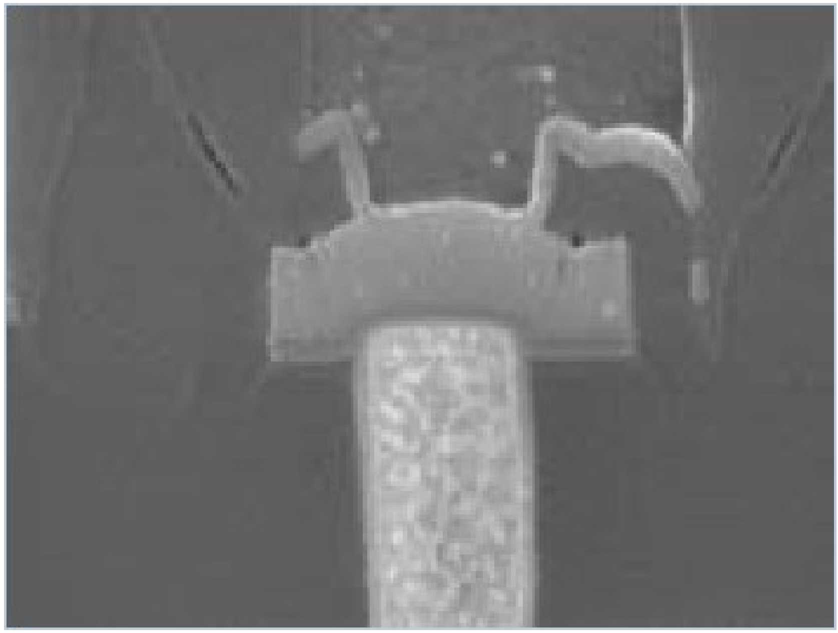
Antifuses

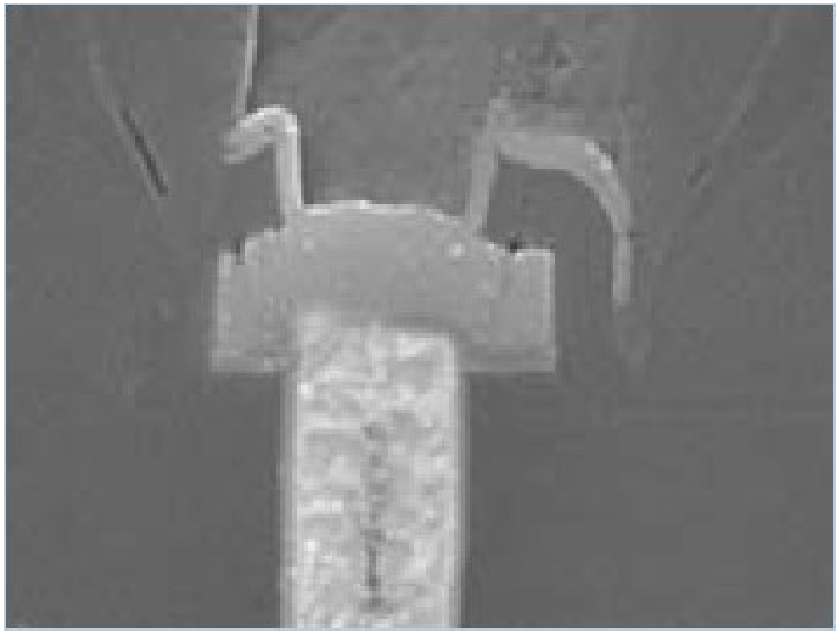
- Opposite of fuse (open until blown)
- Make a connection with electrical signal
  - The current melts a thin insulating layer to form a thin permanent and resistive link.
  - More reliable than breaking a connection (avoids shrapnel)
- Permanently programmed (Non-volatile)





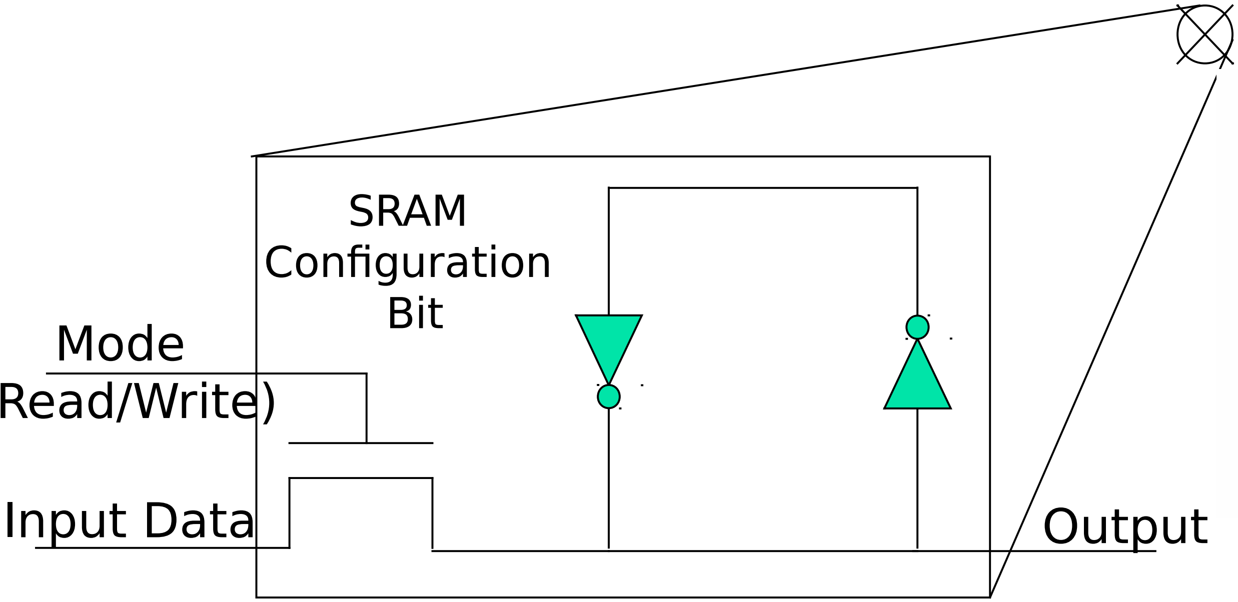
Antifuses

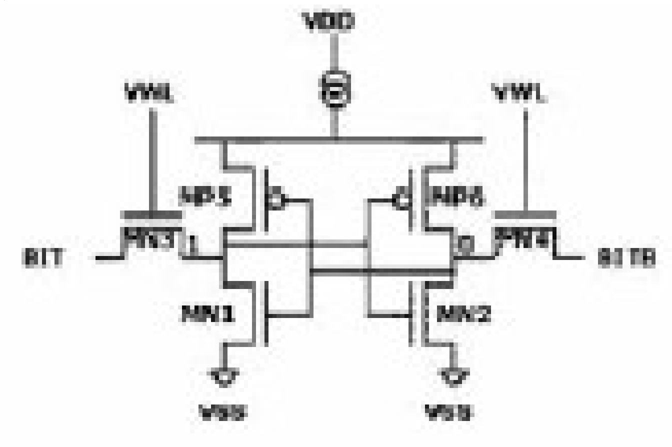




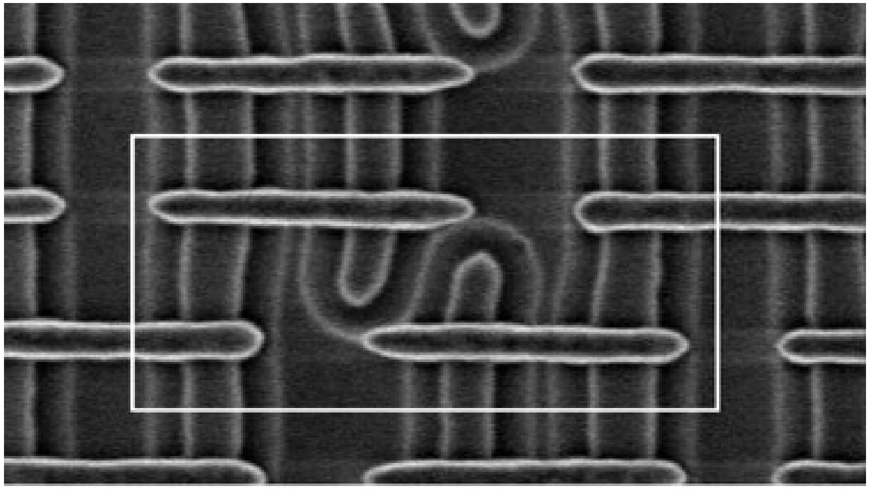
Source: Actel

SRAM



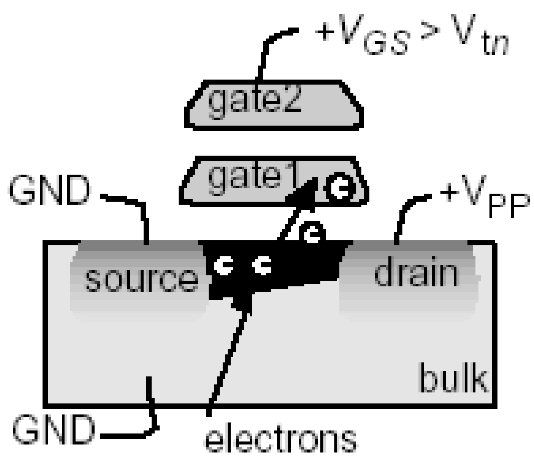


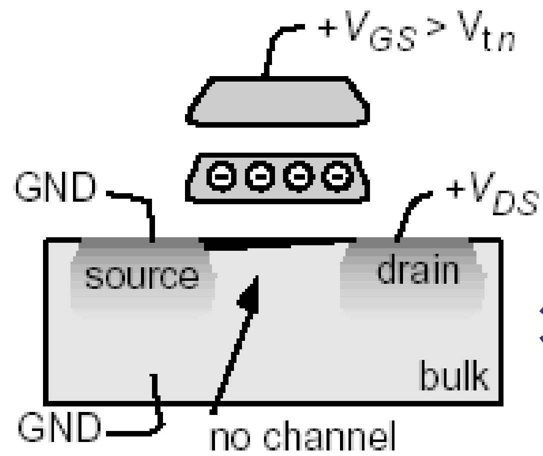
- Volatile
- 6 CMOS transistors
- Standard fabrication process

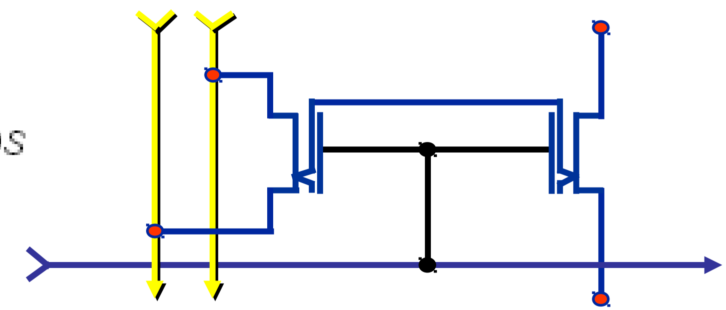


SEM image of a sub-0.49 μm² SRAM cell after gate pattern and etch.

Floating Gate (EPROM/EEPROM/Flash)







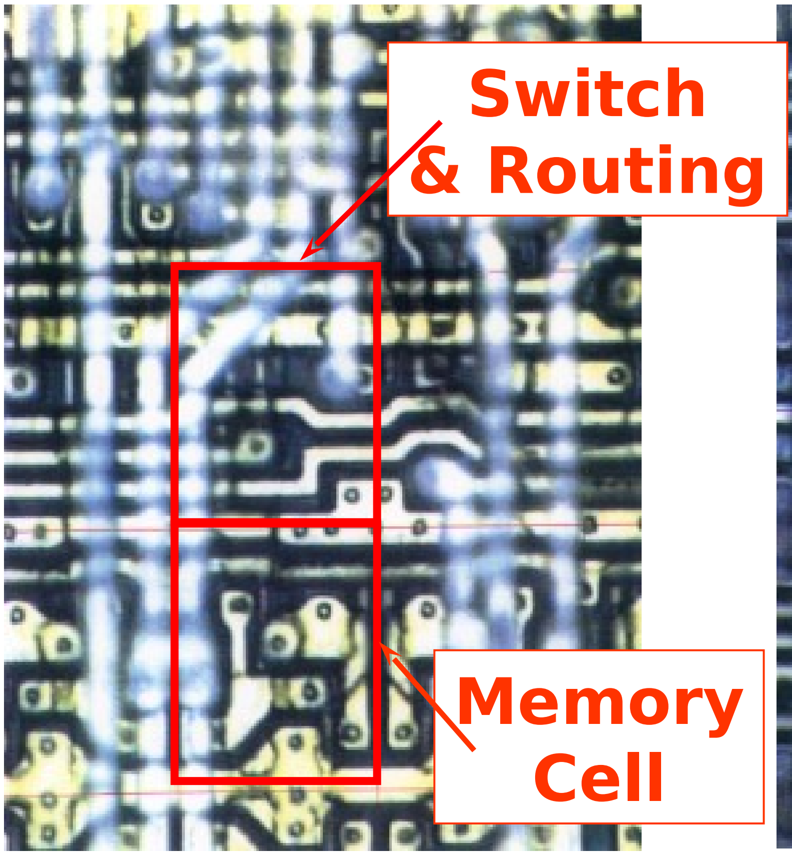
By applying proper programming voltages, electrons “jump” onto floating gate

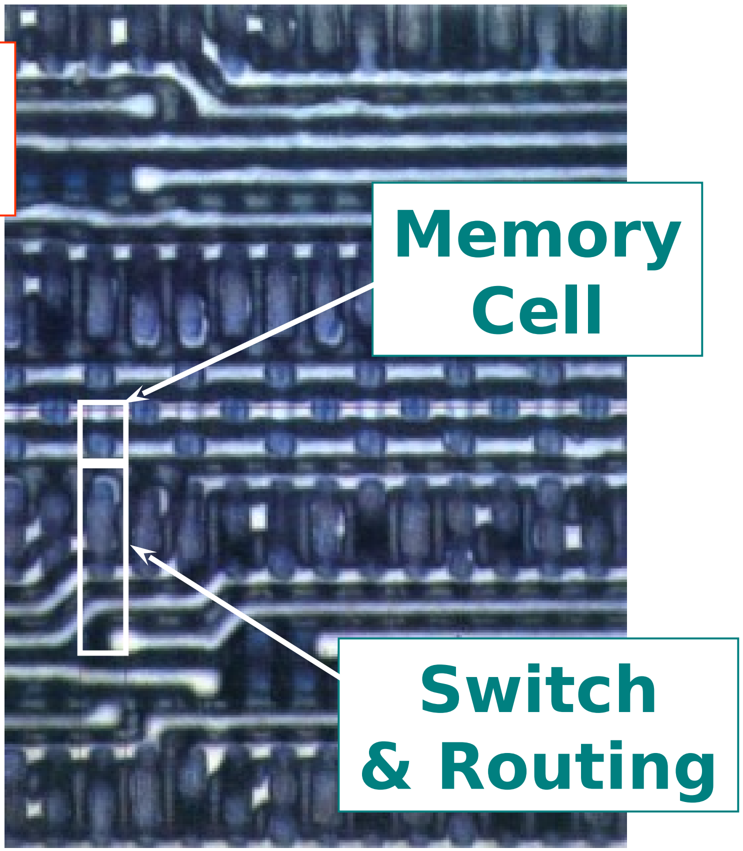
Electrons on gate raise threshold voltage so transistor always off

Flash switch is comprised of two transistors which share a gate

SRAM Versus Flash Switch & Memory Size

SRAM based PLD | **FLASH based PLD**





Programming Technology

	SRAM	Antifuse	Flash
Speed	Medium	Fast	Slow
Power	Poor	Good	Good
Relative Size	1	1/10	1/7
Reprogram	Yes	No	Yes
Size	Large	Small	Moderate

Security

Hardware Security Issues

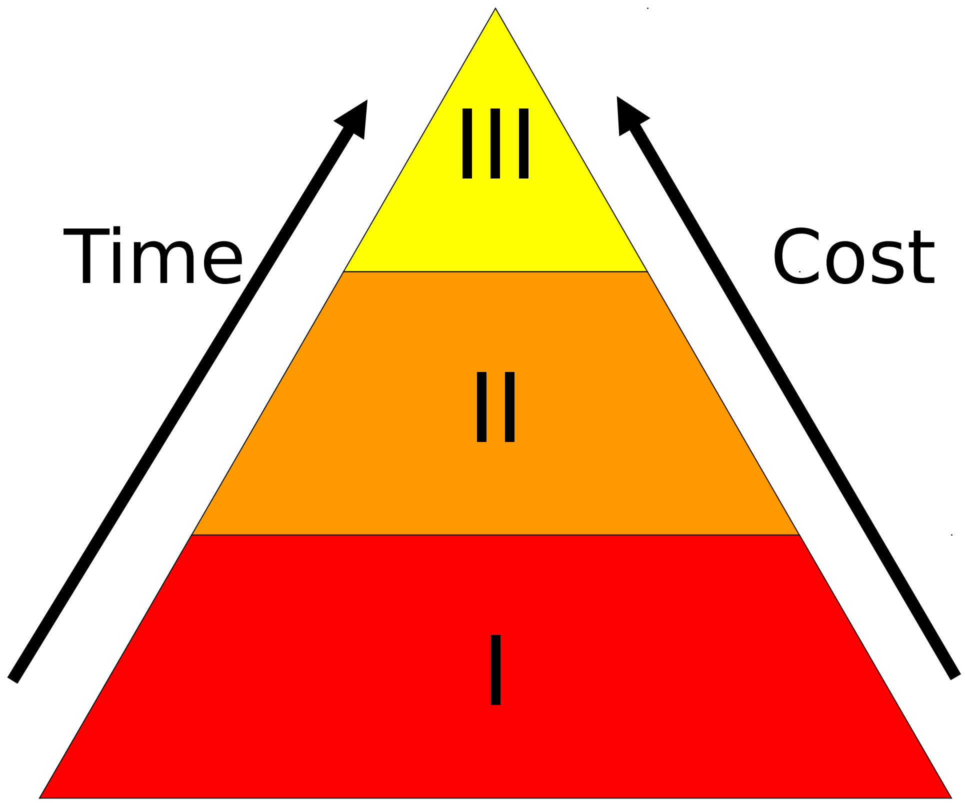
- **Overbuilding** – buying standard parts on open market and making extra illegal products
- **Cloning** – Copying code and duplicating the IP
- **Reverse Engineering** – Reconstructing schematic or netlist to understand and possibly improve and/or disguise the design
- **Denial of Service** – Reprogramming a critical part of the system rendering the entire system inoperable

Hardware Security Attacks

- Non-invasive: Monitoring by external means
  - Brute force key generation
  - Manipulating inputs and observing outputs/system response
  - Probing/copying external code
- Invasive: Decapping and microprobing
  - Focused ion beams
  - Scanning electron microscopes
  - Laser for removal of metalization layers

Levels of Semiconductor Security

- Level I – Not secure, easily compromised with low cost tools (high school kid)
- Level II – Can be broken with time and expensive equipment (commercial enterprise)
- Level III – Can be broken with lost of resources (gov't sponsored lab)



FPGA Attacks

- Black Box Attack
  - Try all possible input combinations, observe outputs
  - Becomes infeasible on complex devices
- Readback Attack
  - Read configuration/state of the FPGA through JTAG or programming interfaces
  - Disable interface, block bitstream readout, embed in secure environment (delete/destroy device if tampered)
- Configuration Cloning
  - Eavesdrop on configuration data stored in external PROM
  - Encrypt the configuration data, decrypt on-chip (store key on-chip)

FPGA Attacks

- Physical Attacks
  - Invasive probing to find chip information
  - Attacks are generally quite expensive (Level II/III)
  - Example Attacks
    - Look for physical changes (“burning”) in memory cells
    - Sectioning silicon and SEM analysis
  - Possible Workarounds
    - Rotate/invert data to avoid burning
    - Reprogram cells randomly initially

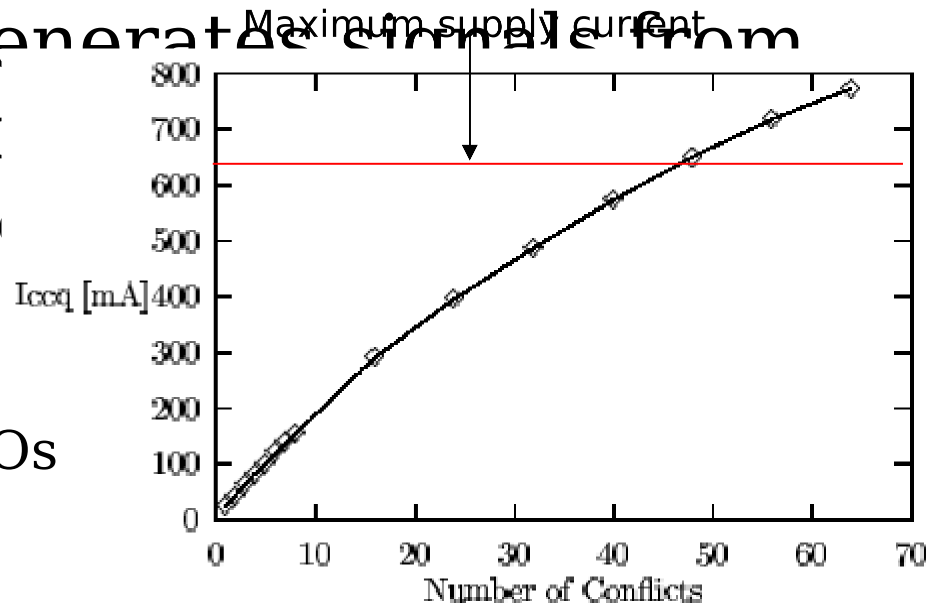
FPGA Attacks

- Side Channel Attacks
  - Observing power consumption, timing, electromagnetic radiation and other unintended information about operation of the FPGA
  - Other applications/cores on FPGA “snoop”
  - Workarounds
    - Vary location of key, encryption/decryption logic
    - Physical separation – hardware guarantees the areas separate
    - Logical separation – software checking for separation

Separation Kernel

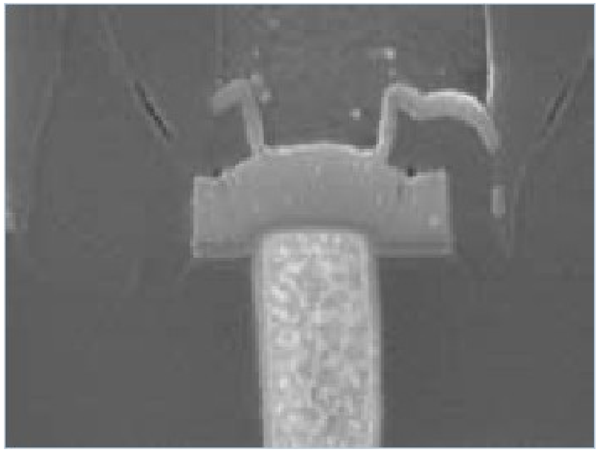
FPGA Virus

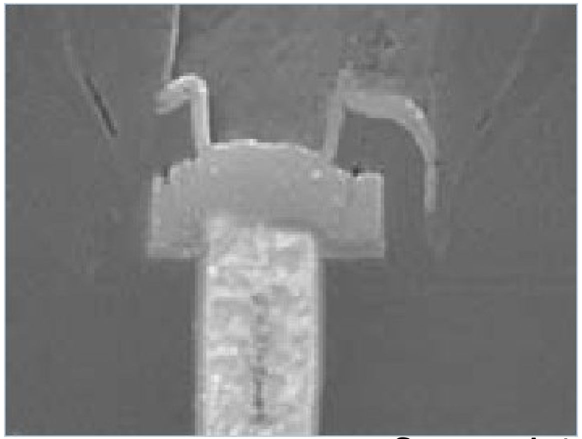
- Bitstream determines functionality of the circuit
- Can be exploited to hang or even destroy system
  - Denial-of-service – generates signals from FPGAs to other devic… sense, hang the devic… them
- Physical destruction
  - Destroy FPGA by creating high currents through intentional logic conflicts
  - Destroy system by creating high currents from FPGA I/Os



Bitstream Verification

Antifuse Security





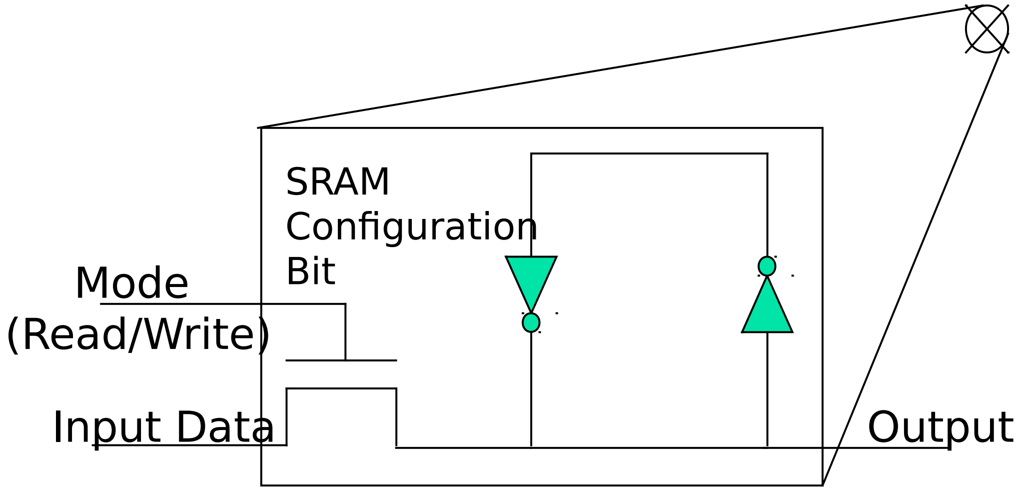
Source: Actel

- All data internal to chip
- No optical change is visible in a programmed antifuse
  - Requires invasive attack - sectioning silicon and SEM analysis
- The larger devices have millions of antifuses
  - Larger devices have 50+ million antifuses
  - Only a small number (< 5%) are typically used
  - Attempts to determine their state physically would take years

Antifuse FPGA: Level II devices

SRAM Security

- SRAM is volatile - configuration data must be loaded each time power is cycled



- Configuration data stored in non-volatile memory external to the FPGA (PROM, FLASH, etc)
- External memories can be physically copied or probed
- Bitstream can be easily capture and duplicated

SRAM FPGA: Level I device

SRAM Security

- Solution: Encrypt bitstream, store key on FPGA
- How to store the key?
  - Non-volatile:
    - On-chip Flash - requires non-standard manufacturing
    - Fuses – unique hardware signatures, cannot be changed
  - Volatile:
    - Key register - must be maintained with a battery
- Must be sure to keep the key safe

SRAM FPGA: Level II device?

Conclusions

- Reconfigurable hardware gives benefits of software and hardware
  - Programmable, but hard to program
  - Performance of hardware (spatial execution)
- Bitstream tells the device what to do – must be protected
- Hardware security issues – cloning, reverse engineering, overbuilding, denial of service, destroying device
- Security issues revolve around protecting bitstream
  - Antifuse, flash – easier to protect (Level II+)
  - SRAM – hard to protect (Level I)

ExPRESS Lab

- ExPRESS - Extensible, Programmable, Reconfigurable Embedded SystemS - http://express.ece.ucsb.edu/
- Students
  - PhD Students – Andrew Brown, Wenrui Gong, Anup Hosangadi, Shahnam Mirzaei, Yan Meng, Gang Wang
  - Undergrad Researchers - Brian DeRenzi, Patrick Lai
- Sponsors: